Lab_05_MultiClass.m

```
%% Machine Learning
% Lab 5: Multi class classification – One VS ALL
% ––– Handwritten Digits –––
%{
For this exercise, you will use logistic regression to
recognize handwritten digits (from 0 to 9). Automated handwritten digit
recognition is widely used today – from recognizing zip codes (postal codes)
on mail envelopes to recognizing amounts written on bank checks.
%}

%% Initialization
clear ; close all; clc

%% Setup the parameters you will use for this part of the exercise
input_layer_size  = 400;  % 20x20 Input Images of Digits
num_labels = 10;          % 10 labels, from 1 to 10
% (note that we have mapped "0" to label 10

%% Loading and Visualizing Data
% Load Training Data
fprintf('Loading and Visualizing Data ...\n')
load('ex3data1.mat'); % training data stored in arrays X, y
m = size(X, 1);

% Randomly select 100 data points to display
rand_indices = randperm(m);
sel = X(rand_indices(1:100), :);

displayData(sel);

fprintf('Program paused. Press enter to continue.\n');
pause;

%% Vectorize Logistic Regression

% Test case for lrCostFunction
fprintf('\nTesting lrCostFunction() with regularization');

w_t = [–2; –1; 1; 2];
X_t = [ones(5,1) reshape(1:15,5,3)/10];
Y_t = ([1;0;1;0;1] >= 0.5);
lambda_t = 3;
[C grad] = lrCostFunction(w_t, X_t, Y_t, lambda_t);

fprintf('\nCost: %f\n', C);
fprintf('Expected cost: 2.534819\n');
fprintf('Gradients:\n');
fprintf(' %f \n', grad);
fprintf('Expected gradients:\n');
fprintf(' 0.146561\n –0.548558\n 0.724722\n 1.398003\n');

fprintf('Program paused. Press enter to continue.\n');
pause;

%% One–vs–All Training
fprintf('\nTraining One–vs–All Logistic Regression...\n');
```

```

lambda = 0.1;
[all_w] = oneVsAll(X, y, num_labels, lambda);

fprintf('Program paused. Press enter to continue.\n');
pause;

%% Predict for One-Vs-All

pred = predictOneVsAll(all_w, X);

fprintf('\nTraining Set Accuracy: %f\n', mean(double(pred == y)) * 100);
```

lrCostFunction.m

```
function [C, grad] = lrCostFunction(w, X, y, lambda)
%Compute cost and gradient for logistic regression with
%regularization
%% Vectorized form

m = length(y); % number of training examples
C = 0;
grad = zeros(size(w));

h = sigmoid(X*w);
% calculate penalty
% excluded the first weight value
w_reg = [0 ; w(2:size(w), :)];
p = lambda*(w_reg'*w_reg)/(2*m);
C = ((-y)'*log(h) - (1-y)'*log(1-h))/m + p;

% calculate grads
grad = (X'*(h - y)+lambda*w_reg)/m;
end
```

oneVsAll.m

```
function [all_w] = oneVsAll(X, y, num_labels, lambda)
%ONEVSALL trains multiple logistic regression classifiers and returns all
%the classifiers in a matrix all_w, where the i-th row of all_w
%corresponds to the classifier for label i
%%
% Some useful variables
m = size(X, 1);
n = size(X, 2);
all_w = zeros(num_labels, n + 1);

% Add ones to the X data matrix
X = [ones(m, 1) X];

for c = 1:num_labels
  init_w = zeros(n+1, 1);
  options = optimset('GradObj', 'on', 'MaxIter', 50);
  [w] = fmincg(@(t)(lrCostFunction(t, X, (y==c), lambda)), init_w, options);
  all_w(c, :) = w';
end

end
```

predictOneVsAll.m

```matlab
function p = predictOneVsAll(all_w, X)
%PREDICT Predict the label for a trained one-vs-all classifier. The labels
%are in the range 1..K, where K = size(all_w, 1).

m = size(X, 1);
num_labels = size(all_w, 1);

p = zeros(size(X, 1), 1);
% Add ones to the X data matrix
X = [ones(m, 1) X];

%%

ps = sigmoid(X*all_w');
[p_max, i_max]=max(ps, [], 2); %max value, max value position
%Max value position corresponds to the actual number
% 10 column - 10 class, if the max value position refers to
% the 10th colum than thats your guess
p = i_max;

end
```